位运算符

1. 左移/右移操作

简单来说，左移/右移就是在二进制bit位上进行操作

- 下面举一个例子：
  - a左移1位

```
int a = 10; //10的二进制表示为0000 1010
a = a << 1; //这里进行左移操作 即二进制位全部向前1位，0001 0100
printf("%d",a);  //于是得到 a 为 20
```

当然，对于人类来说，二进制你很难立马得知这个值是多少。

- 以数学的方式分析：

实际上 对某个数a 进行 "左移", 就是对a * 2的x次方，其中x是以移位的位数来决定的

对某个数a 进行 "右移", 就是对a / 2的x次方，其中x是以移位的位数来决定的

2. 按位 与 操作 （&）

简单来说，按位与操作 就是在二进制bit位上进行比较

- 下面举一个例子：
  - a和b进行比较

```
int a =10,b =11; // 10:0000 1010 //0000 1011
int c = a & b;  //c: 0000 1010
printf("%d",c); //c =10
```

0000 1010

0000 1011 对他们每一位与运算（两个都为1时结果为1，否则为0）

0000 1010 得到c

3. 按位 或 操作（|）

简单来说，按或与操作 就是在二进制bit位上进行比较

- 下面举一个例子：
  - a和b进行比较

```
int a =10,b =11; // 10:0000 1010 //0000 1011
int c = a | b;  //c: 0000 1010
printf("%d",c); //c =11
```

0000 1010

0000 1011 对他们每一位或运算（两个之中只要有一个为1，即为1，否则为0）

0000 1011 得到c

4. 按位 异或 操作（^）

简单来说，按位异或操作 就是在二进制bit位上进行比较

- 下面举一个例子：
  - a和b进行比较

```
int a =10,b =11; // 10:0000 1010 //0000 1011
int c = a | b;  //c: 0000 1010
printf("%d",c); //c =1
```

0000 1010

0000 1011 对他们每一位异或运算（两个数不同即为1，相同为0）

0000 0001 得到c

5. 按位 非 操作（~）

简单来说，按位非操作 就是在二进制bit位上进行取反